

Mikrokomputer K-202



Dokumentacja archiwalna polskiego minikomputera K-202

JĘZYK OPERACYJNY MASZINY – JOM 1

Skanowanie i opracowanie:

Ryszard Zenker

www.zenker.com.pl/k-202

e-mail: k-202@tlen.pl

Puszczykowo – 2008

OŚRODEK DOSKONALENIA KADR TECHNICZNYCH OW/NOT
I
ZAKŁAD DOŚWIADCZALNY MINIKOMPUTERÓW

(Opis wyłącznie do celów szkoleniowych)

JĘZYK OPERACYJNY MASZYNY – JOM 1

Warszawa, wrzesień 1972

S P I S T R E Ś C I

| | Str. |
|--|------|
| 1. PRZEZNACZENIE JĘZYKA OPERACYJNEGO MASZYNY JOM K-202 | 3 |
| 2. DYREKTYWY JOM 1 | 4 |
| 2.1. Uwagi ogólne | 4 |
| 2.1.1. Podawanie dyrektyw | 4 |
| 2.1.2. Kasowanie niezakończonych dyrektyw | 4 |
| 2.1.3. Znaczenie parametru adres 2 | 4 |
| 2.1.4. Śledzenie programu | 4 |
| 2.2. Definicje i oznaczenia | 5 |
| 2.3. Opis dyrektyw JOM 1 | 5 |
| 2.3.1. Wprowadź taśmę binarną | 5 |
| 2.3.2. Wczytaj taśmę binarną | |
| 2.3.3. Startuj od początku wprowadzenia | 5 |
| 2.3.4. Startuj od adresu | 6 |
| 2.3.5. Startuj według stanu IC | 6 |
| 2.3.6. Podaj stan rejestrów | 6 |
| 2.3.7. Podaj stan rejestru | 6 |
| 2.3.8. Podaj stan IC | 6 |
| 2.3.9. Zapisz do rejestru | 6 |
| 2.3.10. Zapisz do IC | 7 |
| 2.3.11. Zapisz do pamięci | 7 |
| 2.3.12. Podaj post-mortem oktalnie | 7 |
| 2.3.13. Podaj post-mortem dziesiętnie | 7 |
| 2.3.14. Podaj post-mortem rozkazowy | 7 |
| 2.3.15. Ustaw powrót do Systemu | 8 |
| 2.3.16. Regeneruj program | 8 |
| 3. ALARMY JOM-4 | 8 |
| 4. ALARMY SYSTEMU OPERACYJNEGO | 9 |

1. PRZEZNACZENIE JĘZYKA OPERACYJNEGO MASZINY JOM K-202

JOM 1 jako program jest częścią Systemu Operacyjnego SOK-1 i służy do dwustronnej wymiany informacji między operatorem i SO K-202. Jako język programowania JOM 1 przedstawia sobą zbiór dyrektyw o charakterze zleceń operatorskich dla Systemu SOK-1.

W dalszej części opisu pojęcie "JOM" będzie używane równolegle w znaczeniu "język programowania" i "PROGRAM- część SO", natomiast aktualne znaczenie będzie wynikało z kontekstu. Komunikacja operator-SOK-1 możliwa jest jedynie przy pomocy dyrektyw JOM-u.

Informacja podawana przez System Operacyjny ma dwojaki charakter; może być to zatem:

- a/ wydruk typu post-mortem; jest to odpowiedź na odpowiednią dyrektywę JOM-u,
- b/ wydruk alarmowy, stanowiący albo sygnalizację błędu w dyrektywie /określany dalej w opisie jako alarm JOM-u/, albo sygnalizację błędu w trakcie wykonywania programu użytkownika /określany dalej jako alarm Systemu Operacyjnego/.

System Operacyjny SOK-1 przyjmuje i wydaje informacje jedynie na monitorze Systemu.

Język Operacyjny Maszyny umożliwia:

- 1/ wyprowadzenie i wprowadzenie taśmy binarnej,
- 2/ wystartowanie z dowolnym programem,
- 3/ zmianę stanu rejestrów, licznika rozkazów IC oraz pamięci operacyjnej PAO,
- 4/ restart,
- 5/ określenie stanu IC i rejestrów,
- 6/ wyprowadzenie post-mortemów oktalnych, dziesiętnych i rozkazowych,
- 7/ śledzenie programu użytkownika.

2. DYREKTYWY JOM 1

2.1. Uwagi ogólne

2.1.1. Podawanie dyrektyw

Podawanie dyrektyw jest możliwe tylko wówczas, gdy JOM sygnalizuje gotowość przyjęcia dyrektywy wydrukiem znaku: na początku nowej linii. JOM generuje nową linię i dwukropkę w następujących przypadkach:

- 1/ po wywołaniu SOK-1 z kluczy bądź deklaracja OS z translatora, ASSK,
- 2/ po wykonaniu rozkazu ex, arg 2. esys.,
- 3/ po alarmie JOM-u,
- 4/ po alarmie systemowym,
- 5/ po wykonaniu poprzedniej dyrektywy.

W przypadkach 1, 2, 4 dodatkowo w poprzedzającym wierszu drukowany jest tekst K-202.

2.1.2. Kasowanie niezakończonych dyrektyw

Możliwe jest kasowanie niezakończonych dyrektyw przez podanie znaku nl.

2.1.3. Znaczenie parametru adres 2

W dyrektywach dwuadresowych parametr adres 2 jest traktowany przez JOM:

- a/ bezpośrednio jako adres końcowy, gdy adres 2 > adres 1,
- b/ jako liczba służąca do obliczenia adresu końcowego, gdy adres 2 jest poprzedzony znakiem + lub -. W tym przypadku adres końcowy = adres 1 + adres 2,
- c/ gdy adres 2 ≤ adres 1 adres końcowy jest równy adres 1.

2.1.4. Śledzenie programu

Dyrektywy T i RT pozwalają na nieniszczący powrót do systemu z wykonywanego programu, wykonanie przy pomocy innych dyrektyw JOM-u dowolnych manipulacji na rejestrach, liczniku rozkazów i pamięci maszyny, a następnie kontynuację programu od miejsca, gdzie został przerwany /lub od dowolnego innego, wskazanego przez aktualny stan IC/ przy pomocy dyrektywy RS.

2.2. Definicja i oznaczenia

Użyte w dalszym opisie pojęcia definiuje się następująco:

$\langle \text{parametr} \rangle ::= \langle \text{liczba krótka} \rangle | \langle \text{parametr} \rangle + \langle \text{liczba krótka} \rangle | \langle \text{parametr} \rangle - \langle \text{liczba krótka} \rangle$
 $\langle \text{ciąg parametrów} \rangle ::= \langle \text{parametr} \rangle ; \langle \text{ciąg parametrów} \rangle \langle \text{parametr} \rangle ;$
 $\langle \text{nr rejestru} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7$
 $\langle \text{adres} \rangle ::= \langle \text{parametr} \rangle$
 $\langle \text{kod dyrektywy} \rangle ::= \text{RB} | \text{WB} | \text{A} | \text{G} | \text{RS} | \text{R} | \text{R} \langle \text{nr rejestru} \rangle | \text{I} | \text{C} | \text{S} | \text{O} | \text{D} | \text{P} | \text{T} | \text{R} | \text{T}$
 $\langle \text{dyrektywa} \rangle ::= \langle \text{kod dyrektywy} \rangle \# \langle \text{kod dyrektywy} \rangle \# \langle \text{parametr} \rangle ; \langle \text{kod dyrektywy} \rangle \# \langle \text{parametr} \rangle ; \langle \text{parametr} \rangle ; \langle \text{kod dyrektywy} \rangle = \langle \text{parametr} \rangle ;$

2.3. Opis dyrektyw JOM 1

2.3.1. Wyprowadź taśmę binarną

Zapis: $\text{WB} \# \langle \text{adres 1} \rangle ; \langle \text{adres 2} \rangle ;$

Znaczenie: Zlecenie wyprowadzenia binarnie zawartości PAO od adresu 1 do adresu 2 włącznie. Wyprowadzana przy pomocy tej dyrektywy taśma binarna opatrzona jest na początku znakiem ? i adresem początku wprowadzania /adres 1/. Na końcu wyprowadzonej taśmy jest umieszczona obliczona suma kontrolna.

Przykłady: $\text{WB} \# 0400; 0500;$
 $\text{WB} \# 0400; + 5;$

2.3.2. Wczytaj taśmę binarną

Zapis: $\text{RB} \#$

Znaczenie: Zlecenie wprowadzenia taśmy binarnej, która była wyprodukowana przy pomocy dyrektywy WB. Na monitorze systemu jest wypisywany adres początku wprowadzania do PAO /adres 1 z dyrektywy WB/.

2.3.3. Startuj od początku wprowadzenia

Zapis: $\text{A}\#$

Znaczenie: Zlecenie startu programu od początku adresu wprowadzania ostatniej wprowadzonej taśmy binarnej. Przed startem rejestry są zerowane.

2.3.4. Startuj od adresu

Zapis: G# <adres>

Znaczenie: Zlecenie startu programu od podanego adresu. Przed startem rejestry są zerowane.

2.3.5. Startuj według stanu IC

Zapis: RS #

Znaczenie: Zlecenie startu programu od adresu ustawionego w IC, bez zmiany stanu rejestrów /tzn. z odtworzeniem zawartości rejestrów/, szczególnie użyteczne jako restart.

2.3.6. Podaj stan rejestrów

Zapis: R #

Znaczenie: Zlecenie wydruku stanu rejestrów R0-R7. Zawartość rejestru R0 drukowana w pierwszym wierszu podana jest w postaci nazw bitów, które są jedynkami. Zawartość R1-R7 podawane są w drugim wierszu jako ciąg liczb oktalnych.

2.3.7. Podaj stan rejestru

Zapis: R <nr rejestru> #

Znaczenie: Zlecenie wydruku stanu wskazanego rejestru. Zawartość rejestru drukowana jest jako liczba oktalna. W przypadku R0 zawartość tego rejestru drukowana jest w postaci opisanej w punkcie 2.3.6.

Przykład: R5 #

2.3.8. Podaj stan IC

Zapis: IC #

Znaczenie: Zlecenie wydruku stanu IC. Zawartość IC drukowana jest jako liczba oktalna.

2.3.9. Zapisz do rejestru

Zapis: R <nr rejestru> = <parametr> ;

Znaczenie: Zlecenie zapisu parametru do wskazanego rejestru.

Przykłady: R1 = 0200;

R2 = 135 + 020;

2.3.10. Zapisz do IC

Zapis: IC = <parametr> ;

Znaczenie: Zlecenie zapisu liczby /sumy liczb/ do IC.

2.3.11. Zapisz do pamięci

Zapis: S \equiv <adres> ; <ciąg parametrów>

Znaczenie: Zlecenie zapisu w PAO poczynając od wskazanego adresu w kolejnych komórkach podanych parametrów. Przejście JOM-u na nasłuch nie jest w tym wypadku sygnalizowane nową linią i :, a samą nową linią. Wyjście z czytania ciągu parametrów następuje po podaniu dowolnej dyrektywy.

Przykłady: S \equiv 0600;

34;

22;

S \equiv 0500;

-1;

2.3.12. Podaj post-mortem oktalnie

Zapis: O \equiv <adres 1> ; <adres 2> ;

Znaczenie: Zlecenie wydruku oktalnego obszaru PAO poczynając od komórki wskazanej przez adres 1 a kończąc na komórce wskazanej przez adres końcowy /patrz 2.1.3./ . Wydruk zawartości każdej z tych komórek poprzedzony jest drukiem adresu komórki; w każdym wierszu drukowane są zatem dwie liczby oktalne, z których pierwsza jest adresem, a druga zawartością komórki.

Przykłady: O \equiv 0500; 0525;

O \equiv 800; + 3;

2.3.13. Podaj post-mortem dziesiętnie

Zapis: D \equiv <adres 1> ; <adres 2> ;

Znaczenie: Zlecenie wydruku dziesiętnego wskazanego obszaru PAO. W każdym wierszu podawany jest adres oktalny komórki i jej zawartość dziesiętnie.

2.3.14. Podaj post-mortem rozkazowy

Zapis: P \equiv <adres 1> ; <adres 2> ;

Znaczenie: Zlecenie wydruku wskazanego obszaru PAO jako ciągu rozkazów. W każdym wierszu podawany jest adres rozkazu /początek rozkazu/ i rozkaz.

2.3.15. Ustaw powrót do Systemu

Zapis: T# <adres> ;

Znaczenie: Zlecenie wpisania pod wskazany adres ekstrakodu ex, 4.esys. Dotychczasowa zawartość dwóch komórek /poczynając od wskazanego adresu/ zostaje przechowana przez System. Użycie dwóch dyrektyw T bez przedzielenia ich dyrektywą RT spowoduje zniszczenie programu; System umożliwia przechowanie tylko dwóch komórek programu na raz. Przechowywane są zawsze komórki programu spod adresu użytego w ostatniej dyrektywie T.

2.3.16. Regeneruj program

Zapis: RT#

Znaczenie: Zlecenie wpisania do komórek wskazanych przez adres występujący w ostatnio użytej dyrektywie T ich poprzedniej zawartości w miejsce rozkazu ex.4 esys.

3. ALARMY JOM-u

Alarmy JOM-u mają postać standardowych tekstów i wypisywane są bezpośrednio po stwierdzeniu przez JOM błędu w dyrektywie. Po wypisaniu odpowiedniego tekstu JOM przechodzi na nasłuch. Poniżej podane zostaną teksty standardowe alarmów JOM-u i ich znaczenia.

- ADDRESS - Odwołanie się do nieistniejącego adresu;
- CONTROL SUM - Obliczenia przy wczytywaniu taśmy binarnej suma kontrolna niezgodna z sumą kontrolną obliczoną przy wyprowadzaniu;
- SYNTAX - Błąd syntaktyczny w dyrektywie;
- CHARACTER - Zły znak w liczbie;
- OVERFLOW - Arytmetyczny nadmiar parametru.

4. ALARMY SYSTEMU OPERACYJNEGO

Alarmy Systemu Operacyjnego mają postać wydruków typu:
ALARM <liczba 1> <liczba 2> <tekst>

Poniżej podane zostaną teksty i ich znaczenia oraz znaczenie parametru "liczba 2" przy poszczególnych alarmach. Parametr "liczba 1" jest adresem pierwszej komórki za rozkazem, próba wykonania którego spowodowała wywołanie alarmu.

- WRONG I/O NUMBER - niewłaściwy symboliczny numer we-wy, użyty w ekstrakodzie,
liczba 2 - symboliczny numer we-wy.
- WRONG PARAMETER - niewłaściwy parametr ekstrakodu,
liczba 2 - wartość parametru.
- WRONG EXTRACODE - użycie nieistniejącego ekstrakodu,
liczba 2 - nr wywołanego ekstrakodu.
- WRONG ADDRESS - błędny adres /nieistniejący/,
liczba 2 - nie istnieje.
- OPERATOR INTERRUPT - przerwane zgłoszenie operatora,
liczba 2 - nie istnieje.
- WRONG INSTRUCTION - rozkaz o nieistniejącym kodzie,
liczba 2 - zawartość komórki o adresie o 1 mniejszym od liczby 1.
- FINISH - wystąpił ekstrakod ESYS,
liczba 2 - argument 2 ekstrakodu ESYS.
Gdy arg 2 = 0 następuje przejście do JOM-u bez sygnalizacji alarmowej.
- F.P. OVERFLOW - nadmiar zmiennoprzecinkowy,
liczba 2 - nie istnieje.
- NOT CONNECTED TERMINAL - nie przyłączone urządzenie we-wy,
liczba 2 - symboliczny nr urządzenia we-wy.